STAT 237
Linear Regression

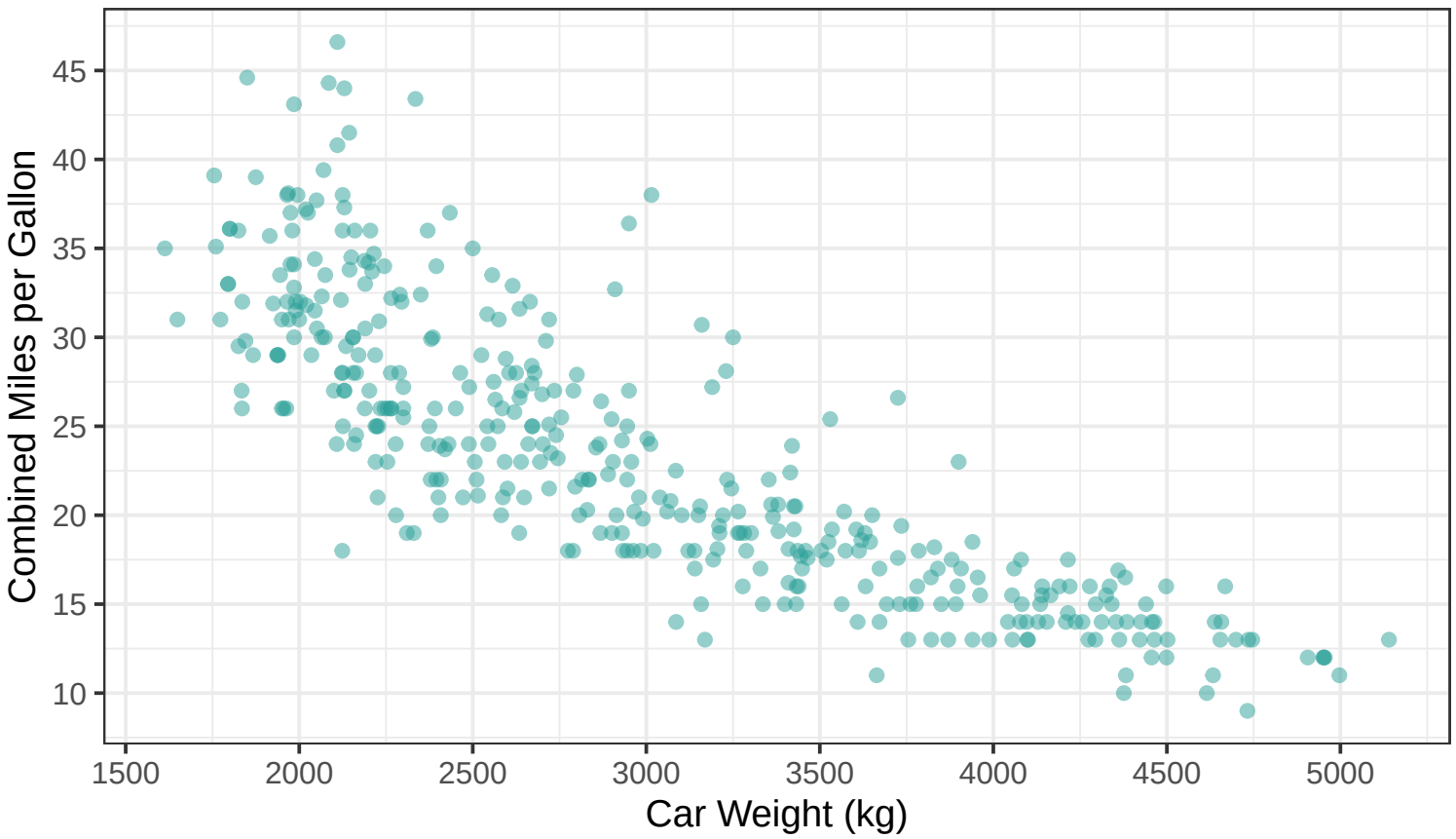
May 16-20, 2022

Colin Reimer Dawson

Outline

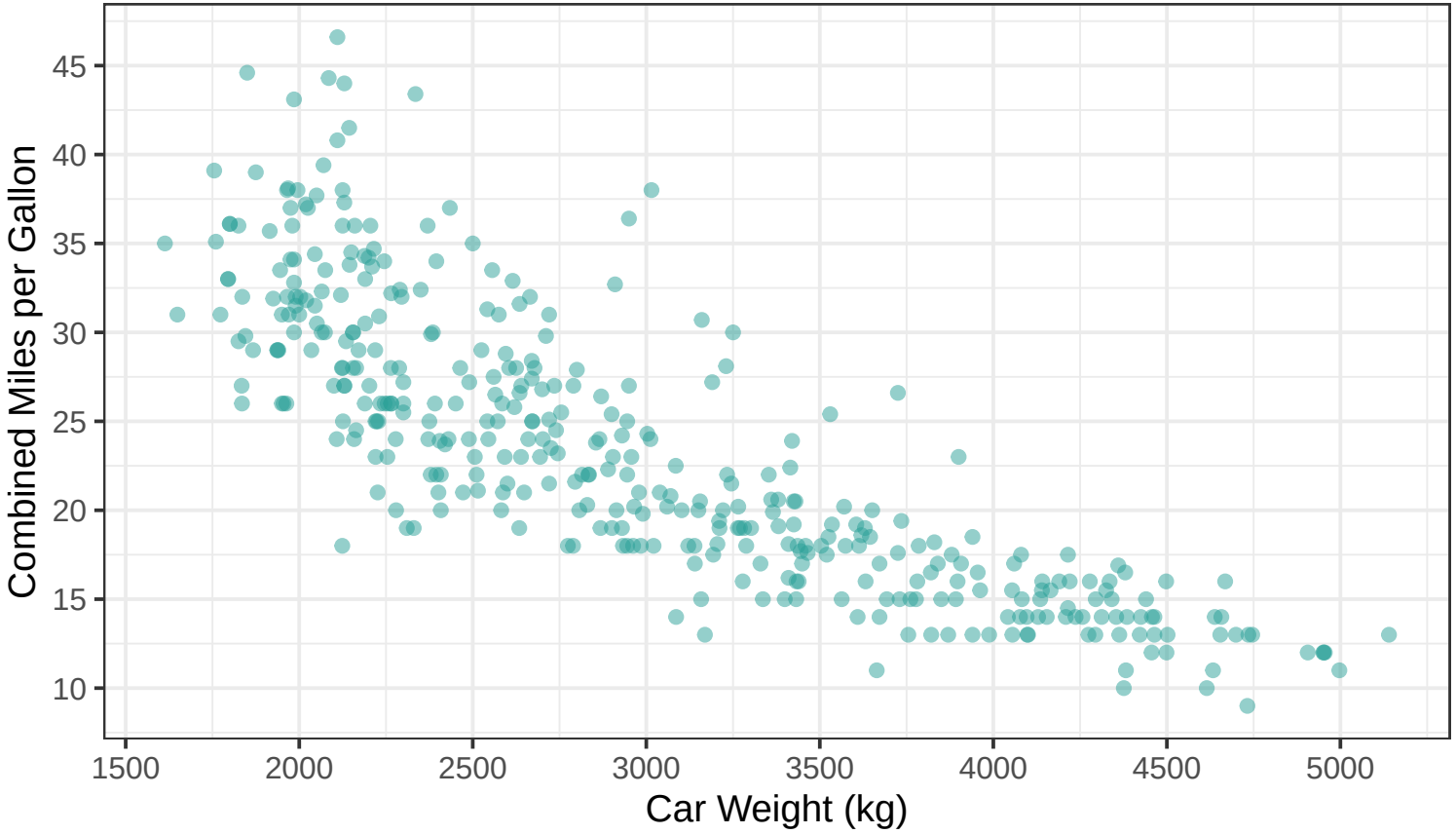
Regression

- Use a **feature vector**, $\mathbf{x}$ to help predict a **quantitative target variable**, y



Regression

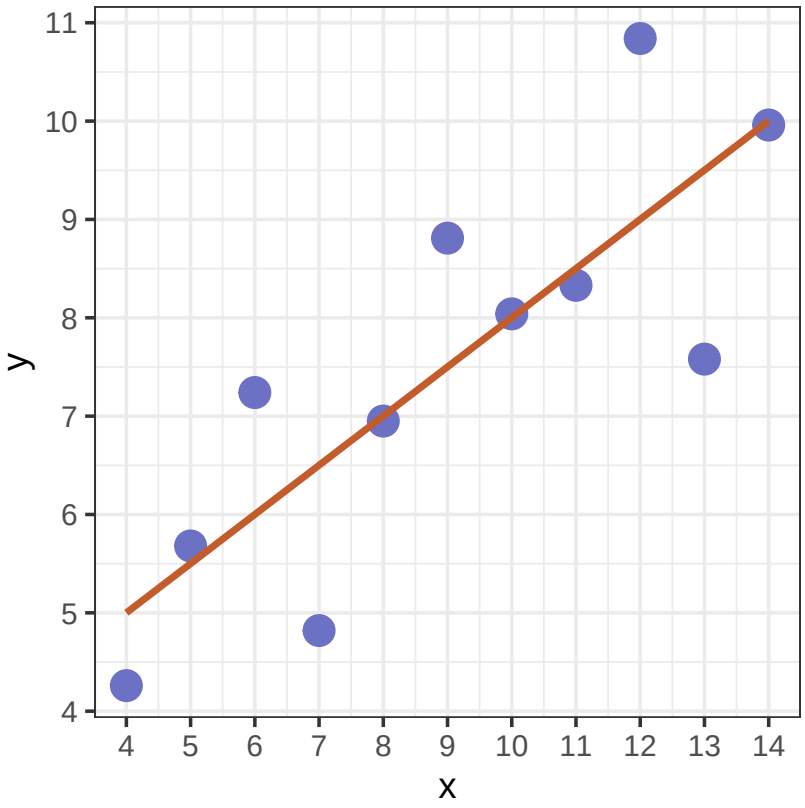
- Use a **feature vector**, $\mathbf{x}$ to help predict a **quantitative target variable**, y
- Goal: use training data $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}, n = 1, \ldots, N$ to learn to predict y_{new} given future $\mathbf{x}_{\text{new}}$: $\hat{y} = f(\mathbf{x})$.



Single Input Feature: Simple Linear Regression

- Simplest case: **one input feature** (1D feature vector)
- Examples:
  - Use height to predict blood pressure
  - Use economic indicators to predict stock prices
  - Use biomarkers to predict disease progression

A Simple Linear Model

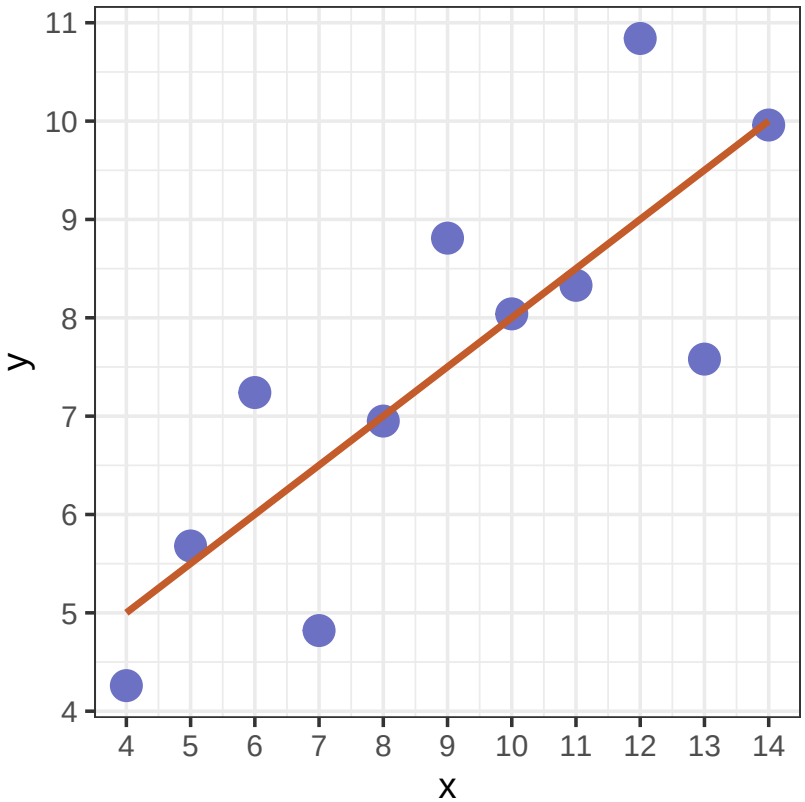


- One of the simplest models is a straight line:

$$\hat{y}_n = f(x_n) = \beta_0 + \beta_1 x_n$$

The values β_0 and β_1 are **parameters** of the model.

A Simple Linear Model



- One of the simplest models is a straight line:

  $$\hat{y}_n = f(x_n) = \beta_0 + \beta_1 x_n$$

  The values β_0 and β_1 are **parameters** of the model.
- How should we choose values for the parameters?

Outline

Minimizing Prediction Error

Classical Regression: Choose parameters so as to minimize a **loss function**, $\mathcal{L}$, which measures the discrepancy between the predicted and actual values of y

Minimizing Prediction Error

Classical Regression: Choose parameters so as to minimize a **loss function**, $\mathcal{L}$, which measures the discrepancy between the predicted and actual values of y

Ordinary Least Squares (OLS) Regression: Use sum of the squares of these differences (called residuals) as the loss function

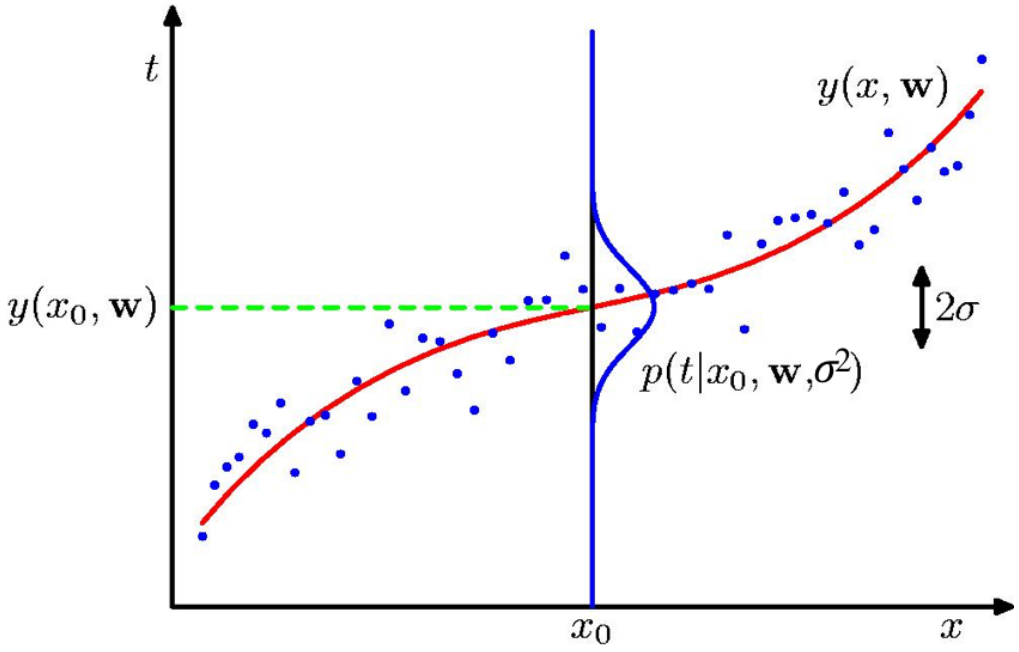
$$\mathcal{L}(\mathbf{y}, f(\mathbf{X})) := \sum_n (y_n - f(\mathbf{x}_n))^2$$

A Generative Linear Model

If each observation is associated with a random ε_n term, then we have a **generative model**:

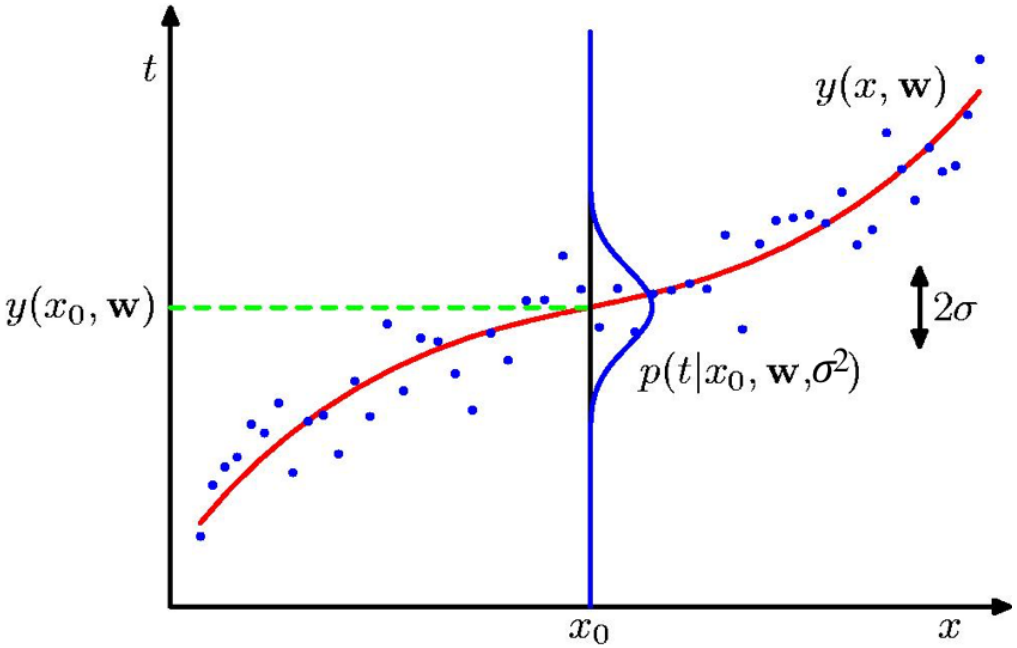
$$y_n = f(x_n) + \varepsilon_n$$

where ε_n is a **random error** term.



Regression with I.I.D. Normal "noise"

The classic case is when the ε_i are **independent, and identically distributed** as $\mathcal{N}(0, \sigma^2)$ random variables.



Outline

Generative Model

- A general form of a generative linear regression model

$$y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_D x_{nD} + \varepsilon_n, \quad \{\varepsilon_n\}_{n=1}^N \sim \mathcal{N}(0, \sigma^2)$$

Generative Model

- A general form of a generative linear regression model

$$y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_D x_{nD} + \varepsilon_n, \quad \{\varepsilon_n\}_{n=1}^N \sim \mathcal{N}(0, \sigma^2)$$

- Parameters of the generative model are

$$\boldsymbol{\beta} := (\beta_0, \beta_1, \ldots, \beta_D)$$
$$\sigma^2 \quad \text{(or a one-to-one function of it)}$$

Generative Model

- A general form of a generative linear regression model

$$y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_D x_{nD} + \varepsilon_n, \quad \{\varepsilon_n\}_{n=1}^N \sim \mathcal{N}(0, \sigma^2)$$

- Parameters of the generative model are

$$\begin{aligned} \boldsymbol{\beta} &:= (\beta_0, \beta_1, \ldots, \beta_D) \\ \sigma^2 & \quad \text{(or a one-to-one function of it)} \end{aligned}$$

- To do Bayesian inference, need a prior on $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2)$

Generative Model

- A general form of a generative linear regression model

$$y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_D x_{nD} + \varepsilon_n, \quad \{\varepsilon_n\}_{n=1}^N \sim \mathcal{N}(0, \sigma^2)$$

- Parameters of the generative model are

$$\begin{aligned} \boldsymbol{\beta} &:= (\beta_0, \beta_1, \ldots, \beta_D) \\ \sigma^2 & \quad \text{(or a one-to-one function of it)} \end{aligned}$$

- To do Bayesian inference, need a prior on $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2)$
- A Gamma distribution is a **conjugate prior** for $1/\sigma^2$

Generative Model

- A general form of a generative linear regression model

$$y_n = \beta_0 + \beta_1 x_{n1} + \beta_2 x_{n2} + \cdots + \beta_D x_{nD} + \varepsilon_n, \quad \{\varepsilon_n\}_{n=1}^N \sim \mathcal{N}(0, \sigma^2)$$

- Parameters of the generative model are

$$\begin{aligned} &\boldsymbol{\beta} := (\beta_0, \beta_1, \ldots, \beta_D) \\ &\sigma^2 \quad \text{(or a one-to-one function of it)} \end{aligned}$$

- To do Bayesian inference, need a prior on $\boldsymbol{\theta} = (\boldsymbol{\beta}, \sigma^2)$
- A Gamma distribution is a **conjugate prior** for $1/\sigma^2$
- Conditional on σ^2, a conjugate prior on $\mathbf{w}$ is a **multivariate Normal distribution**

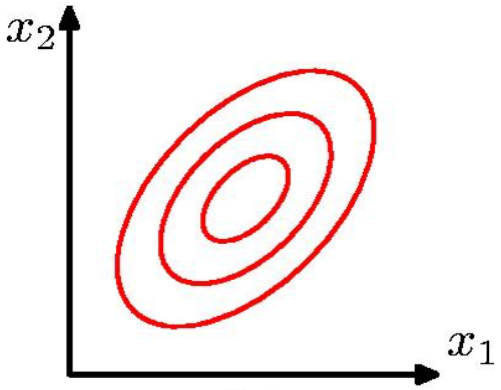
$$\boldsymbol{\beta} \mid \boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$$

where $\boldsymbol{\mu}$ is the **prior mean vector** and $\boldsymbol{\Sigma}_0$ is the **prior covariance matrix**

Aside: Multivariate Normal Random Vector

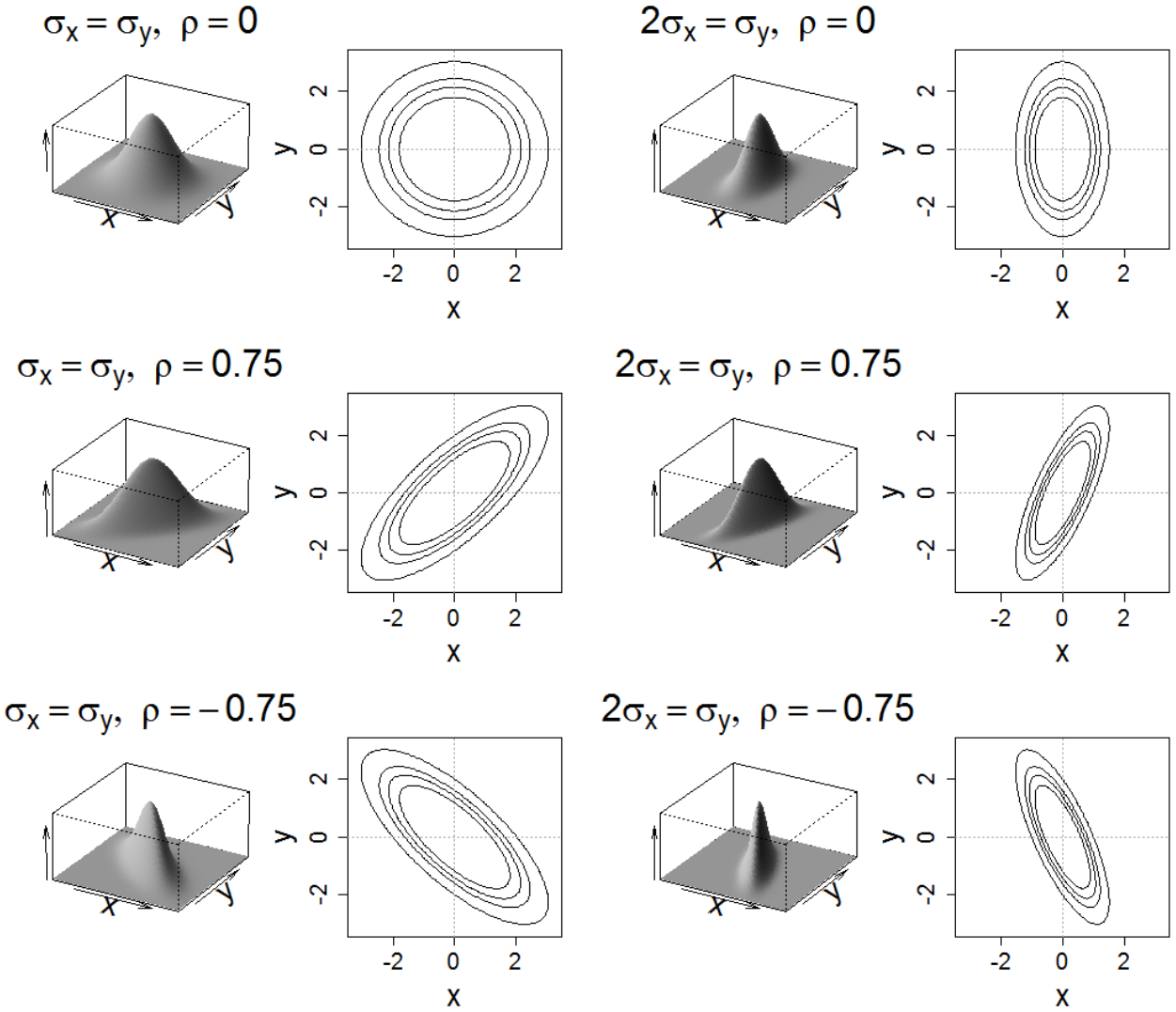
The random vector, $\mathbf{x}$ has a D-dimensional **multivariate normal distribution** with **mean vector** $\boldsymbol{\mu}$ and **covariance matrix** $\boldsymbol{\Sigma}$ if its density (in $\mathbb{R}^D$) is

$$p(\mathbf{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-D/2} |\boldsymbol{\Sigma}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\}$$



where $|\boldsymbol{\Sigma}|$ is the **determinant** of the matrix (a scalar proportional to the size of the contour ellipse containing a fixed probability)

Varying the Covariance Matrix



Bayesian Regression as a Graphical Model

Outline

Pulse Rates

```
library(Stat2Data)
data(Pulse)
PulseModified <- Pulse %>%
    mutate(
        Smoke = factor(Smoke),
        Male  = factor(1 - Sex),
        BMI   = Wgt / Hgt^2 * 708)
sample(PulseModified, size = 10) %>%
    select(Active, Rest, Smoke, Male, BMI)

     Active Rest Smoke Male      BMI
165     60   55     0    1 21.85185
103     92   58     1    1 27.64970
222     88   57     0    1 28.19252
180    139   72     1    0 19.97884
155    102   84     0    0 25.92773
136    114   74     0    0 23.65783
28      61   53     0    1 25.28571
151     82   68     0    1 26.63194
181     86   58     0    1 26.73061
212     89   60     0    0 25.04891
```

Active Pulse Rate by Smoker Status

- A linear regression model to the `Active` pulse rate variable, with the binary `Smoke` variable as the sole predictor
- Coefficients are optimized using Ordinary Least Squares

```
model_smoke <- lm(Active ~ Smoke, data = PulseModified)
model_smoke %>% coef() %>% round(digits = 2)

     (Intercept)      Smoke1
           90.52        6.94
```

What is the model here?

Active Pulse Rate by Smoker Status

- A linear regression model to the `Active` pulse rate variable, with the binary `Smoke` variable as the sole predictor
- Coefficients are optimized using Ordinary Least Squares

```
model_smoke <- lm(Active ~ Smoke, data = PulseModified)
model_smoke %>% coef() %>% round(digits = 2)

     (Intercept)      Smoke1
           90.52        6.94
```

What is the model here?
What does the coefficient for `Smoke` represent?

Combining Quantitative and Indicator Variables

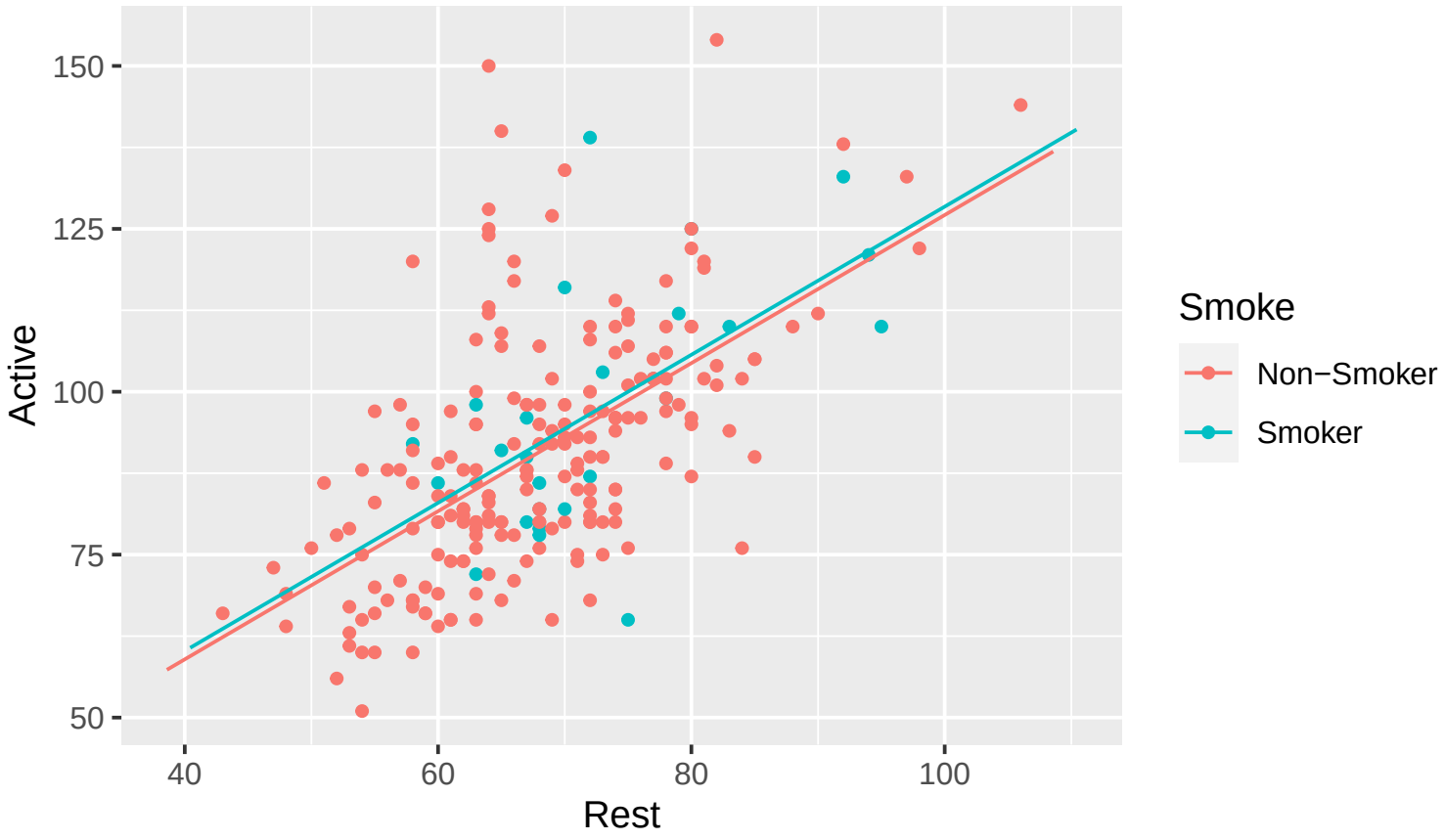
```
model_smoke_rest <-
    lm(Active ~ Rest + Smoke, data = PulseModified)
model_smoke_rest %>% coef() %>% round(digits = 2)

     (Intercept)        Rest      Smoke1
           13.48        1.14        1.29
```

$$\texttt{Active} = 13.48 + 1.14 \cdot \texttt{Rest} + 1.29 \cdot \texttt{Smoke}$$

Now what does the `Smoke` coefficient tell us?

```r
## CAUTION: don't try to use this with multiple quantitative
## predictors; it won't make sense
plotModel(model_smoke_rest) +
    scale_color_discrete(
        name = "Smoke",
        labels = c("0" = "Non-Smoker", "1" = "Smoker"))
```



One Model, Two Prediction Equations

$$\texttt{Active} = 13.48 + 1.14 \cdot \texttt{Rest} + 1.29 \cdot \texttt{Smoke}$$

One Model, Two Prediction Equations

$$\texttt{Active} = 13.48 + 1.14 \cdot \texttt{Rest} + 1.29 \cdot \texttt{Smoke}$$

Non-Smokers: $\texttt{Active} = 13.48 + 1.14 \cdot \texttt{Rest}$

Smokers: $\texttt{Active} = (13.48 + 1.29) + 1.14 \cdot \texttt{Rest}$

```r
model_smoke_rest %>% coef() %>% round(digits = 2)

     (Intercept)        Rest      Smoke1
           13.48        1.14        1.29
```

Non-Parallel Lines

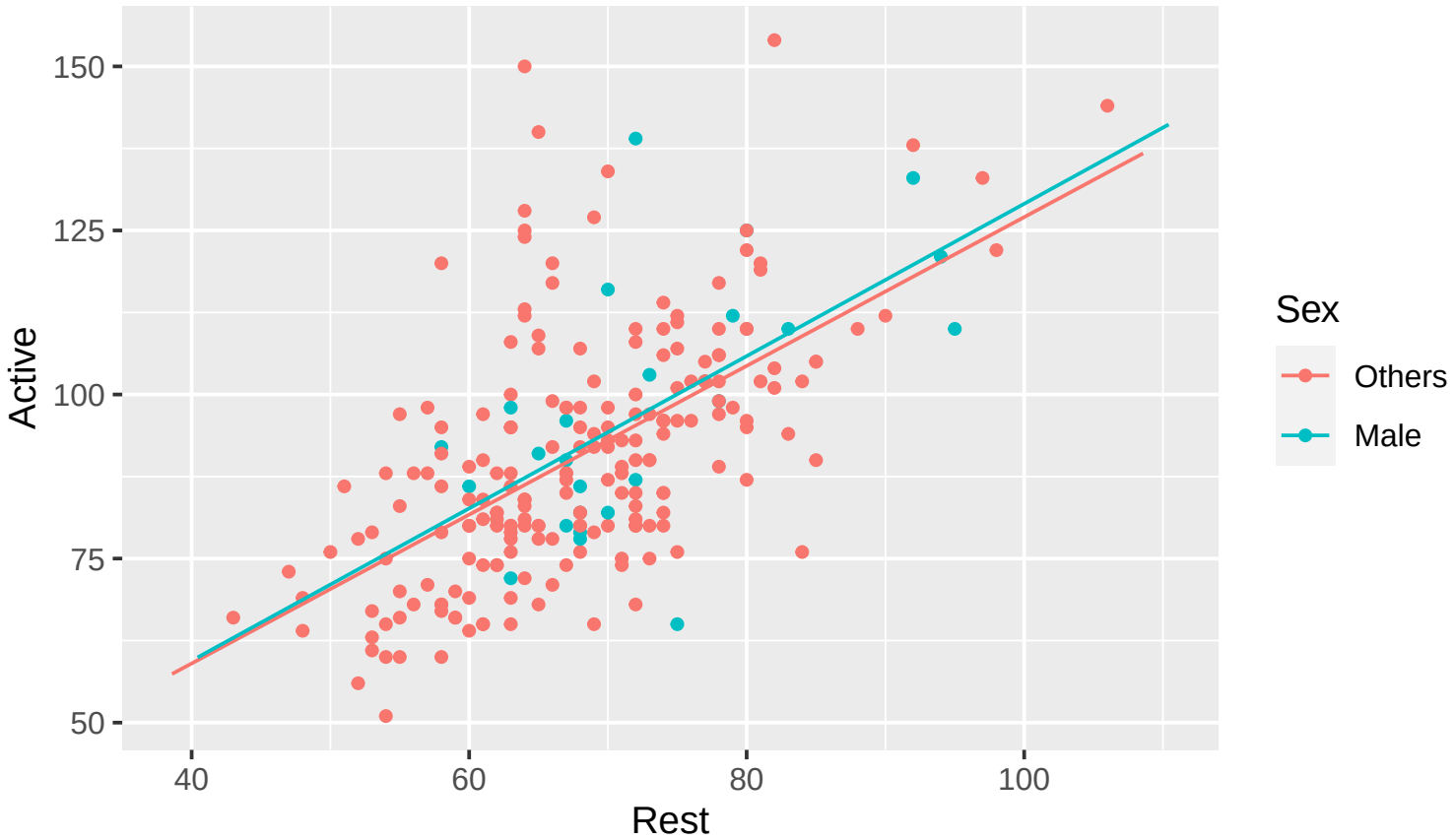
```
model_r_s_rs <- lm(Active ~ Rest + Smoke + Rest:Smoke, data = PulseModified)
model_r_s_rs %>% coef() %>% round(digits = 2)

     (Intercept)        Rest      Smoke1 Rest:Smoke1
           13.68        1.13       -0.66        0.03
```

$$\texttt{Active} = 13.68 + 1.13 \cdot \texttt{Rest} - 0.66 \cdot \texttt{Smoke} + 0.027 \cdot \texttt{Rest} \cdot \texttt{Smoke}$$

Now what does the `Smoke` coefficient tell us? The last coefficient?

```r
## CAUTION: don't try to use this with multiple quantitative
## predictors; it won't make sense
plotModel(model_r_s_rs) +
    scale_color_discrete(
        name = "Sex",
        labels = c("0" = "Others", "1" = "Male"))
```



Non-Parallel Lines

- `Smoke` coefficient is the difference in intercepts
- the **interaction term** is the difference in slopes

$$\texttt{Active} = 13.68 + 1.13 \cdot \texttt{Rest} - 0.66 \cdot \texttt{Smoke} + 0.027 \cdot \texttt{Rest} \cdot \texttt{Smoke}$$

Non-Parallel Lines

- `Smoke` coefficient is the difference in intercepts
- the **interaction term** is the difference in slopes

$$\texttt{Active} = 13.68 + 1.13 \cdot \texttt{Rest} - 0.66 \cdot \texttt{Smoke} + 0.027 \cdot \texttt{Rest} \cdot \texttt{Smoke}$$

Non-Smokers: $\texttt{Active} = 13.68 + 1.13 \cdot \texttt{Rest}$

Smokers: $\texttt{Active} = (13.68 - 0.66) + (1.13 + 0.027) \cdot \texttt{Rest}$

```r
model_r_s_rs %>% coef() %>% round(digits = 2)

     (Intercept)        Rest      Smoke1 Rest:Smoke1
           13.68        1.13       -0.66        0.03
```

Centering a Predictor

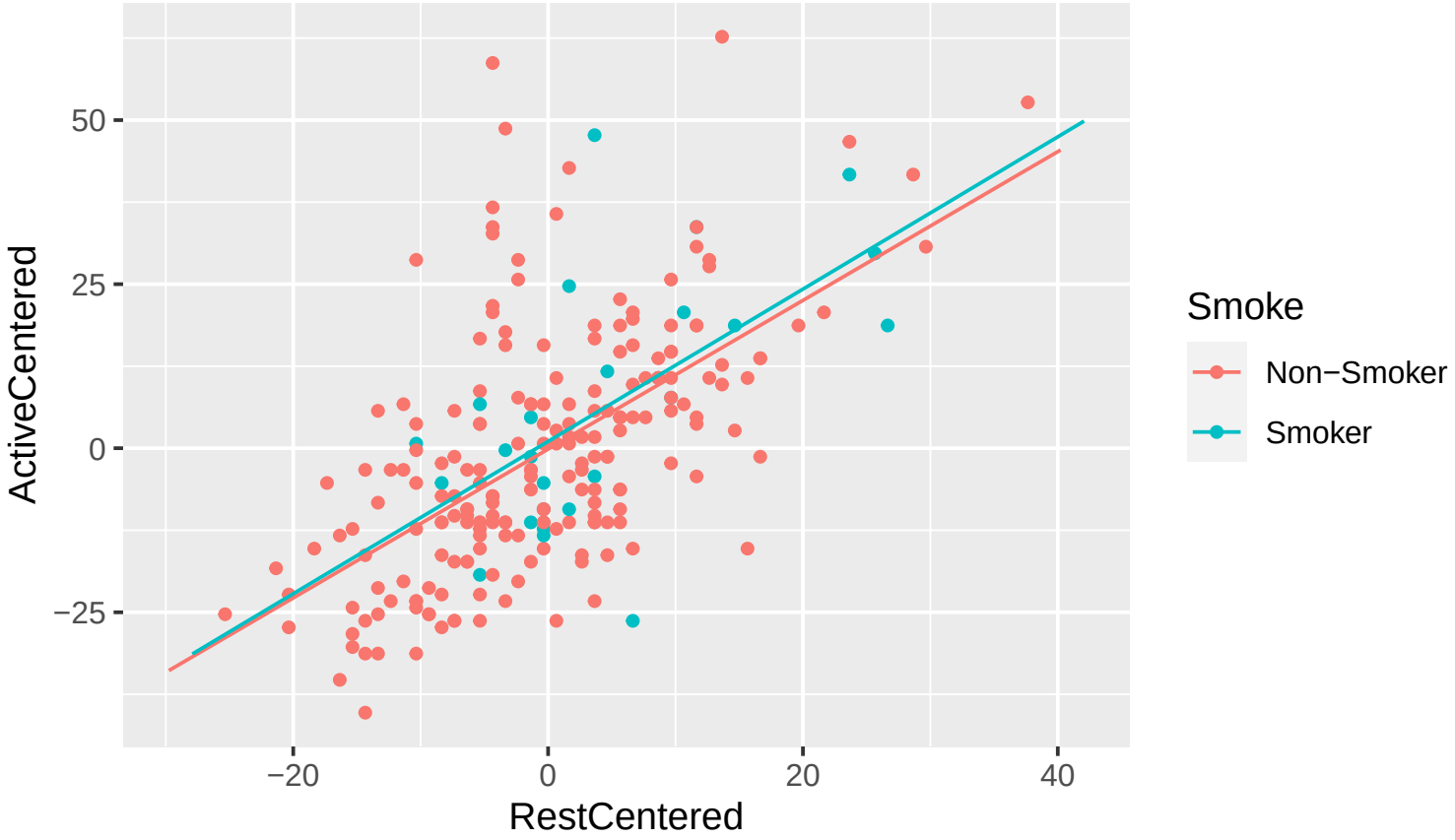
```
PulseCentered <- PulseModified %>%
    mutate(
        RestCentered   = Rest   - mean(Rest),
        ActiveCentered = Active - mean(Active))
model_r_s_rs <-
    lm(ActiveCentered ~ RestCentered + Smoke + RestCentered:Smoke,
       data = PulseCentered)
model_r_s_rs %>% coef() %>% round(digits = 2)

             (Intercept)        RestCentered               Smoke1 RestCentered:S
                   -0.15                1.13                 1.18
```

$$\widehat{\texttt{ActiveCentered}} = -0.15 + 1.13 \cdot \texttt{RestCentered} + 1.18 \cdot \texttt{Smoke}$$
$$0.027 \cdot \texttt{RestCentered} \cdot \texttt{Smoke}$$

Now what does the coefficient in front of `Smoke` tell us?

```r
plotModel(model_r_s_rs) +
    scale_color_discrete(
        name   = "Smoke",
        labels = c("0" = "Non-Smoker", "1" = "Smoker"))
```



Outline

- The dataset `NCbirths` has records from a sample of 1450 births in North Carolina in 2001.
- A question of interest is how birth weights (`BirthWeightOz`) might differ according by race
- The variable `MomRace` codes the mother's "race" as Black, Latinx, "Other"[1], or White.

[1] "Other" encompasses American Indian, Chinese, Japanese, Hawaiian, Filipino, and Other Asian or Pacific Islander

Reference Coding

- We have a variable (`MomRace`) which is **categorical, but not binary**

Reference Coding

- We have a variable (`MomRace`) which is **categorical, but not binary**
- To fit a regression model with this predictor, we need to break this up into multiple **indicator variables**

Reference Coding

- We have a variable (`MomRace`) which is **categorical, but not binary**
- To fit a regression model with this predictor, we need to break this up into multiple **indicator variables**
- In this model, one category is chosen as the **reference category**

Reference Coding

- We have a variable (`MomRace`) which is **categorical, but not binary**
- To fit a regression model with this predictor, we need to break this up into multiple **indicator variables**
- In this model, one category is chosen as the **reference category**
- Each **other level** has an indicator variable which is 1 for cases in that category

Reference Coding

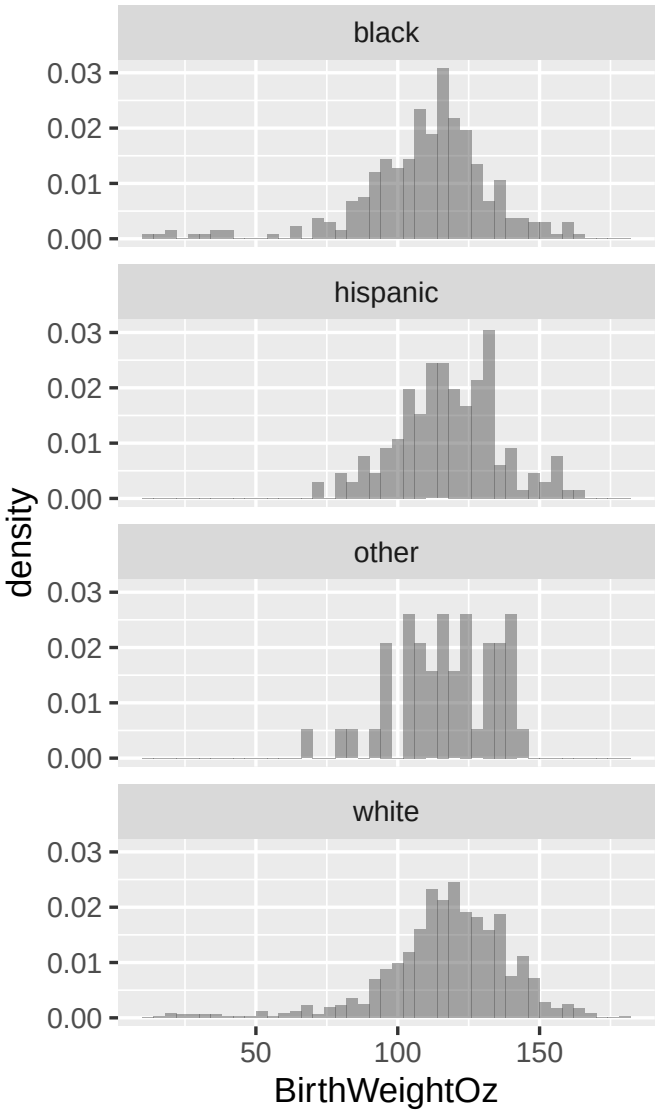
- We have a variable (`MomRace`) which is **categorical, but not binary**
- To fit a regression model with this predictor, we need to break this up into multiple **indicator variables**
- In this model, one category is chosen as the **reference category**
- Each **other level** has an indicator variable which is 1 for cases in that category
- Cases in the reference category have **zero for every indicator**

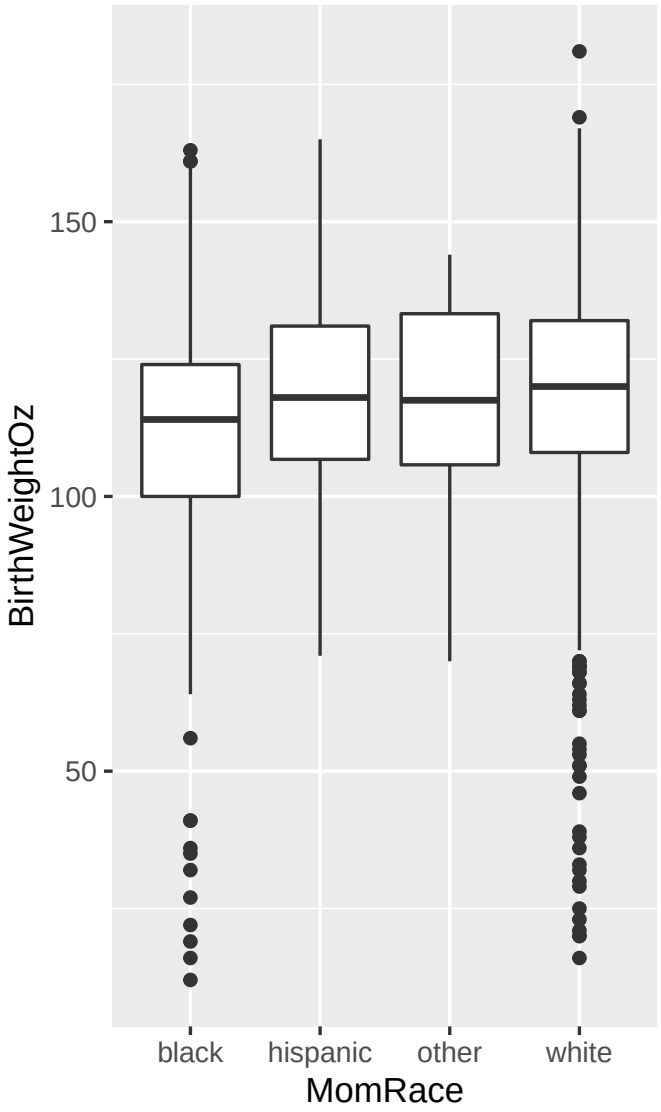
Two Representations

Case	BirthWeightOz	MomRace
1	125	white
2	108	hispanic
3	139	other
4	118	black
5	113	hispanic

Case	BirthWeightOz	white	hispanic	other
1	125	1	0	0
2	108	0	1	0
3	139	0	0	1
4	118	0	0	0
5	113	0	1	0

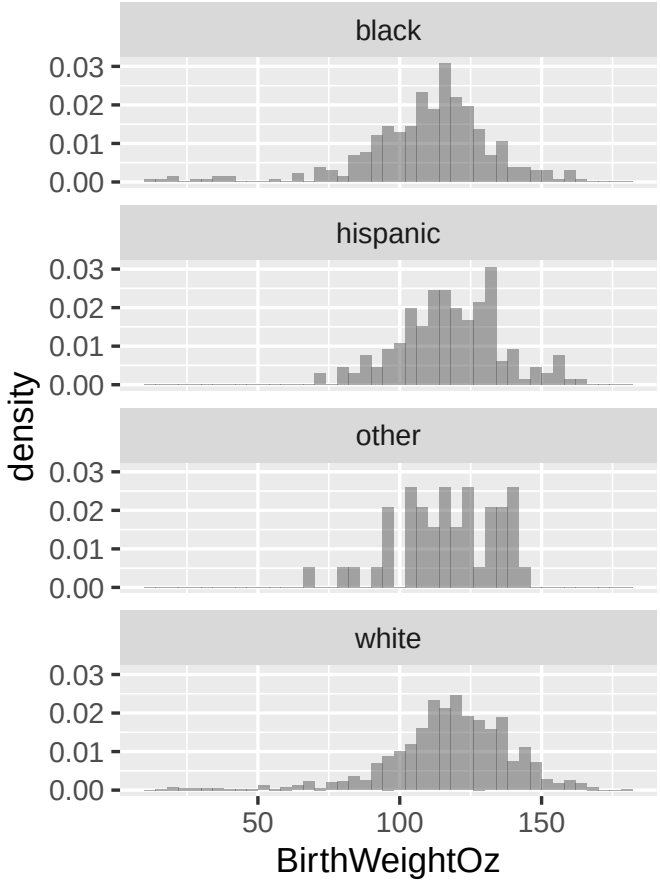
The Data

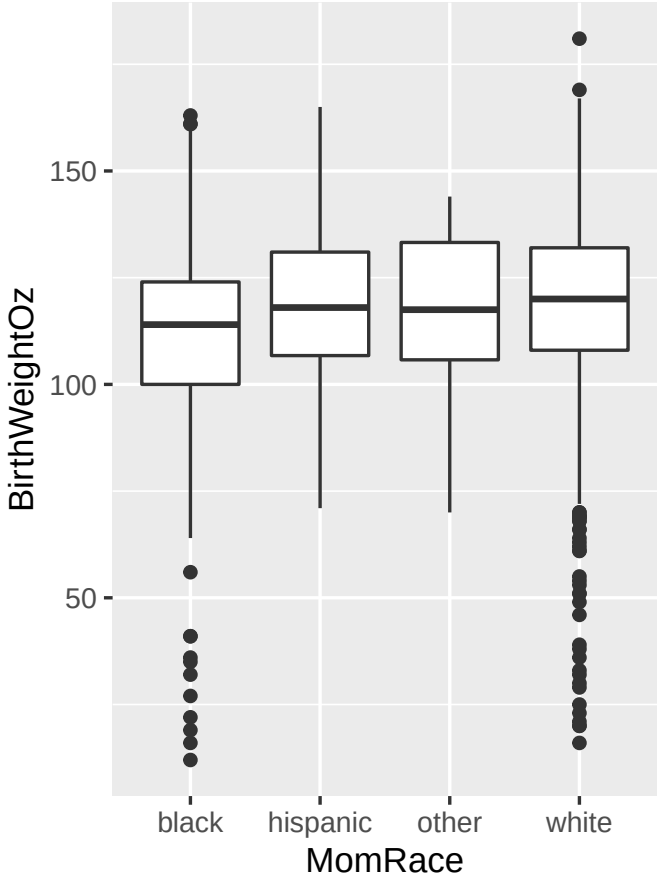

black
hispanic
other
white
0.03
0.02
0.01
0.00
density
50
100
150
BirthWeightOz


150
100
50
BirthWeightOz
black
hispanic
other
white
MomRace

For Reference: R Code

```
library(Stat2Data); library(mosaic); data(NCbirths)
gf_dhistogram(~BirthWeightOz, data = NCbirths, binwidth = 4) +
    facet_wrap(~MomRace, nrow = 4, ncol = 1)
gf_boxplot(BirthWeightOz ~ MomRace, data = NCbirths)
```





Prediction Equation

```
RaceModel <- lm(BirthWeightOz ~ MomRace, data = NCbirths)
RaceModel %>% coef() %>% round(2)

         (Intercept) MomRacehispanic    MomRaceother    MomRacewhite
              110.56            7.96            6.58            7.31
```

$$\widehat{\texttt{BirthWeightOz}} = 117.87 + 7.96 \cdot \texttt{hispanic} + 6.58 \cdot \texttt{other} + 7.31 \cdot \texttt{white}$$

The indicator variables are 1 when the mother identifies with the race in question, and zero otherwise.

- Q: What does each coefficient tell us about race and birth weights? (Assume that each mother picks exactly one category to identify with.)

Prediction Equations by Group

```
RaceModel %>% coef() %>% round(2)

         (Intercept) MomRacehispanic    MomRaceother    MomRacewhite
              110.56            7.96            6.58            7.31
```

$$\widehat{\texttt{BirthWeightOz}} = 117.87 + 7.96 \cdot \texttt{hispanic} + 6.58 \cdot \texttt{other} + 7.31 \cdot \texttt{white}$$

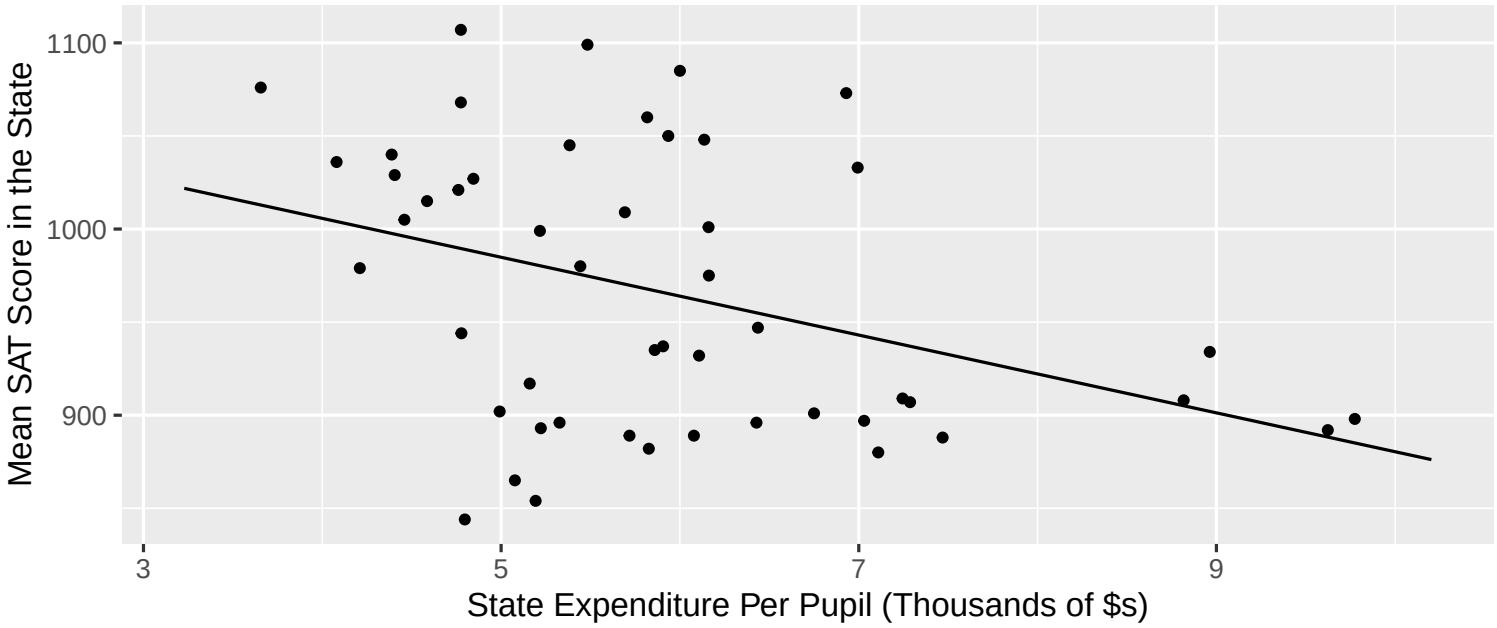
$$\widehat{\texttt{BirthWeightOz}}_i = \begin{cases} 110.56 & \text{if } \texttt{MomRace}_\texttt{i} = \texttt{black} \\ 110.56 + 7.96 & \text{if } \texttt{MomRace}_\texttt{i} = \texttt{hispanic} \\ 110.56 + 6.58 & \text{if } \texttt{MomRace}_\texttt{i} = \texttt{other} \\ 110.56 + 7.31 & \text{if } \texttt{MomRace}_\texttt{i} = \texttt{white} \end{cases}$$

Outline

State Education Spending and SAT Scores

```
library(mosaic); data(SAT)
```

```
m_expend <- lm(sat ~ expend, data = SAT)
plotModel(m_expend) +
    xlab("State Expenditure Per Pupil (Thousands of $s)") +
    ylab("Mean SAT Score in the State")
```



State Education Spending and SAT Scores

```
m_expend %>% summary() %>% coef() %>% round(3)

                  Estimate Std. Error t value Pr(>|t|)
     (Intercept) 1089.294     44.390  24.539    0.000
     expend       -20.892      7.328  -2.851    0.006
```

Question: What should we make of this?

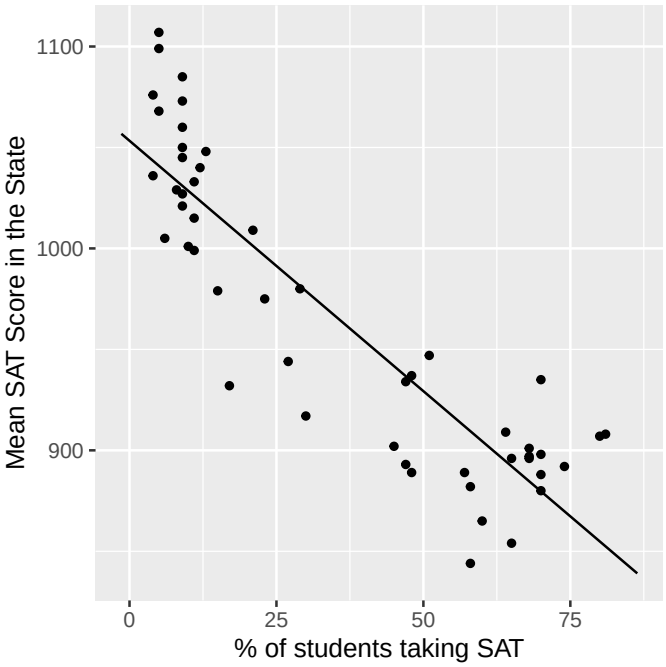
SAT Scores and Participation Rate

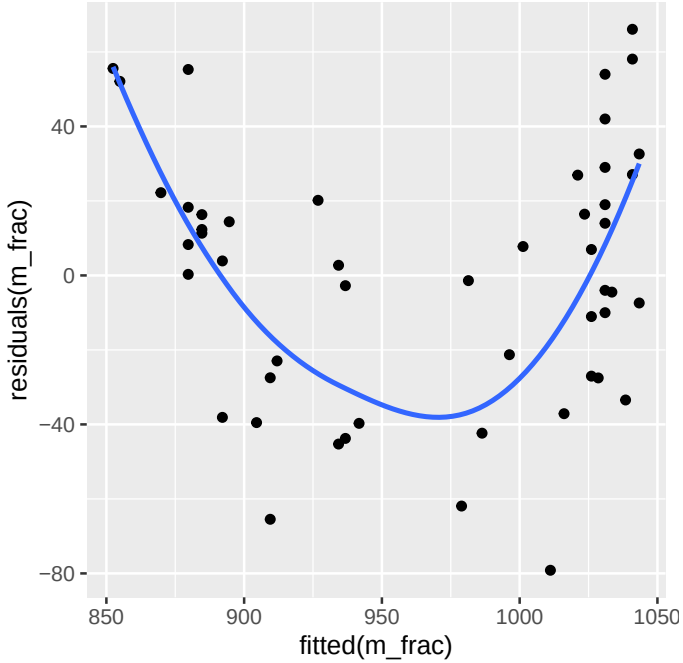
```
m_frac <- lm(sat ~ frac, data = SAT) ## frac = % taking SAT
```

```
plotModel(m_frac) +
    xlab("% of students taking SAT")
    ylab("Mean SAT Score in the Stat
```

```
gf_point(
    residuals(m_frac) ~
        fitted(m_frac)) %>%
gf_smooth()

    `geom_smooth()` using
method = 'loess'
```





Polynomial Regression

We can create "new" predictors from old, e.g.:

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \cdots + \beta_D X_i^D + \varepsilon_i$$

$$D = \begin{cases} 1, & \text{linear} \\ 2, & \text{quadratic} \\ 3, & \text{cubic} \\ etc. & \end{cases}$$

R: Two Equivalent Methods

Method 1: Inline transformation (note use of `I()`)

```
m_frac_quadratic <- lm(sat ~ frac + I(frac^2), data = SAT)
```

```
m_frac_quadratic


     Call:
     lm(formula = sat ~ frac + I(frac^2), data = SAT)

     Coefficients:
     (Intercept)         frac    I(frac^2)
      1094.09787     -6.52850      0.05242
```

R: Two Equivalent Methods

Method 2: Using `poly()` to generate polynomials (note `raw = TRUE`)

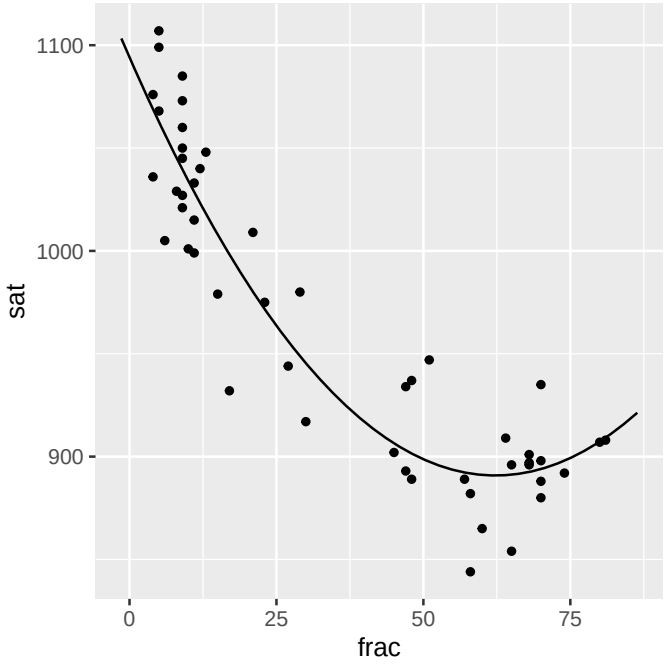
```
m_frac_quadratic2 <- lm(
    sat ~ poly(frac, degree = 2, raw = TRUE),
    data = SAT)
```

```
     Call:
     lm(formula = sat ~ poly(frac, degree = 2, raw = TRUE), data = SAT)

     Coefficients:
                                (Intercept)  poly(frac, degree = 2, raw = TRUE)1
                                 1094.09787                             -6.52850
     poly(frac, degree = 2, raw = TRUE)2
                                  0.05242
```

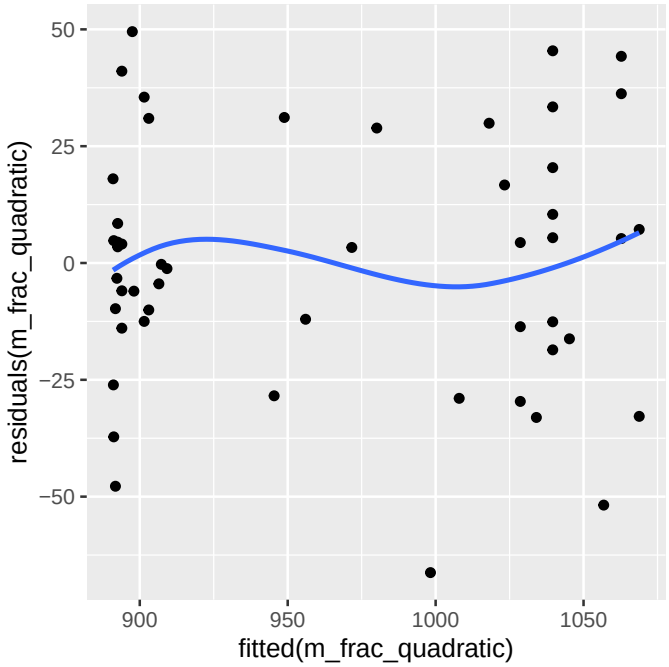
Example: State SAT Scores

```
plotModel(m_frac_quadratic)
```




```
gf_point(
    residuals(m_frac_quadratic) ~
        fitted(m_frac_quadratic)) %>%
    gf_smooth()

     `geom_smooth()` using
method = 'loess'
```



ASSESS: Do we need the quadratic term?

```
m_frac_quadratic %>% summary() %>% coef()

                    Estimate  Std. Error    t value     Pr(>|t|)
(Intercept) 1094.09786793 9.643906886 113.449651 5.567889e-59
frac          -6.52849528 0.730624639  -8.935498 1.063033e-11
I(frac^2)      0.05241712 0.009271252   5.653727 8.961681e-07
```

Questions:

1. What nested models are being compared in the t-test of the quadratic coefficient?
2. What nested models are being compared in the t-test of the linear coefficient?

Selecting Polynomial Order

- When comparing polynomial models, **it is generally inadvisable to have "gaps" in the powers you include**
- Doing this without a solid domain-knowledge reason quite often yields violations of regression conditions.
- Don't remove lower order terms even if nonsignificant!

Outline

Spending After Controlling for Participation Rate

```
cor(expend ~ frac, data = SAT)

     [1] 0.5926274
```

```
m_frac_expend <- lm(sat ~ frac + expend, data = SAT)
```

```
m_frac_expend %>% summary() %>% coef() %>% round(3)

                 Estimate Std. Error t value Pr(>|t|)
     (Intercept)  993.832     21.833  45.519    0.000
     frac          -2.851      0.215 -13.253    0.000
     expend        12.287      4.224   2.909    0.006
```

Question: How can we interpret the coefficient for `expend` here?

Quadratic Control for Participation Rate

```
m_frac_quad_expend <- lm(sat ~ frac + I(frac^2) + expend, data = SAT)
```

```
m_frac_quad_expend %>% summary() %>% coef() %>% round(3)

                 Estimate Std. Error t value Pr(>|t|)
     (Intercept) 1051.887     20.825  50.511    0.000
     frac          -6.381      0.704  -9.068    0.000
     I(frac^2)      0.047      0.009   5.175    0.000
     expend         7.914      3.498   2.262    0.028
```

Question: How can we interpret the coefficient for `expend` here?

Interaction Terms and Second-Order Models

Consider the model:

$$\widehat{\texttt{sat}}_i = \beta_0 + \beta_{\texttt{e}} \cdot \texttt{expend}_i + \beta_{\texttt{f}} \cdot \texttt{frac}_i + \beta_{\texttt{ef}} \cdot \texttt{expend}_i \cdot \texttt{frac}_i$$

How can we interpret $\beta_{\texttt{ef}}$?

Interaction Terms and Second-Order Models

Consider the model:

$$\widehat{\texttt{sat}}_i = \beta_0 + \beta_{\texttt{e}} \cdot \texttt{expend}_i + \beta_{\texttt{f}} \cdot \texttt{frac}_i + \beta_{\texttt{ef}} \cdot \texttt{expend}_i \cdot \texttt{frac}_i$$

How can we interpret $\beta_{\texttt{ef}}$?

$$\widehat{\texttt{sat}}_i = (\beta_0 + \beta_{\texttt{f}}\texttt{frac}_i) + (\beta_{\texttt{f}} + \beta_{\texttt{ef}}\texttt{frac}_i)\texttt{expend}_i$$

Interaction Terms and Second-Order Models

Consider the model:

$$\widehat{\texttt{sat}}_i = \beta_0 + \beta_{\texttt{e}} \cdot \texttt{expend}_i + \beta_{\texttt{f}} \cdot \texttt{frac}_i + \beta_{\texttt{ef}} \cdot \texttt{expend}_i \cdot \texttt{frac}_i$$

How can we interpret $\beta_{\texttt{ef}}$?

$$\widehat{\texttt{sat}}_i = (\beta_0 + \beta_{\texttt{f}}\texttt{frac}_i) + (\beta_{\texttt{f}} + \beta_{\texttt{ef}}\texttt{frac}_i)\texttt{expend}_i$$

$\beta_{\texttt{ef}}$ represents change in slope relating `sat` to `expend` for **each unit increase** in `frac` (or vice versa)

Interaction Model

```
m_frac_expend_interaction <-
    lm(sat ~ frac + expend + frac:expend, data = SAT)
m_frac_expend_interaction %>%
    summary() %>%
    coef() %>%
    round(3)

             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1057.121     42.040  25.146    0.000
frac          -4.232      0.818  -5.175    0.000
expend         0.629      7.846   0.080    0.936
frac:expend    0.237      0.135   1.748    0.087
```

Interaction Visualization

Demo

The Economic Value of a College Degree

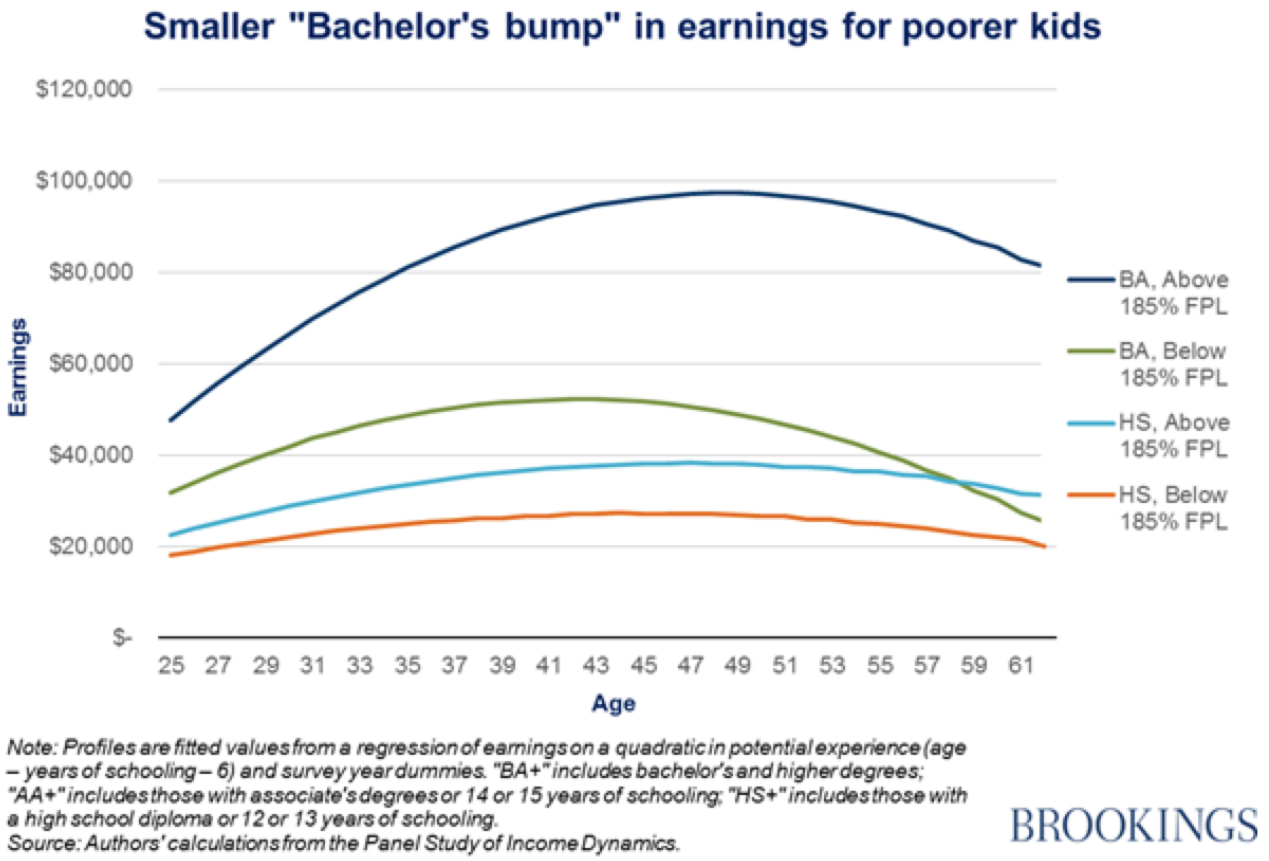


Figure: Source: `http://www.pbs.org/newshour/making-sense/if-you-grew-up-poor-your-college-degree-may-be-worth-less/`